Naval Research Laboratory

Washington, DC 20375-5000

NRL/FR/6521—92-9392

A Local Interpolator Derived From the Discrete Fourier Transform

ROBERT L. LUCKE

Advanced Concepts Branch
Optical Sciences Division

April 30, 1992

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 30, 1992	Final

4. TITLE AND SUBTITLE

A Local Interpolator Derived From the Discrete Fourier Transform

5. FUNDING NUMBERS

PR - RA11W52
WU - 2558
PE - 62111N

6. AUTHOR(S)

Robert L. Lucke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5001

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/6521—92-9392

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Technology
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The discrete Fourier transform is used to obtain an analog of the Nyquist reconstruction formula for a limited number of data points. The resulting interpolator, when applied to the center of a sliding window (rather than applied globally, as is usually done with Fourier-related techniques), can extend good interpolation performance to lower data sampling rates than is possible with polynomial interpolators.

14. SUBJECT TERMS

Interpolator
Discrete Fourier transform

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

CONTENTS

A LOCAL INTERPOLATOR DERIVED FROM THE DISCRETE FOURIER TRANSFORM

1. INTRODUCTION

Interpolating between measured data points is a common signal processing problem. Standard interpolation methods [1], which use polynominal fits to the sampled data, perform well, but even better performance is desirable for critical applications. In a frame-differencing signal processor, for example, the difference between two successive pictures is taken to "zero out" stationary backgrounds and render faint moving targets detectable. Since the samples from the two pictures (usually) do not coincide, values must be interpolated between samples of the second picture to compare to the first. The more accurately this can be done, the less clutter will occur in the differenced image.

In the ideal case, when there is a large number of samples and the Nyquist sampling criterion is satisfied, the Nyquist reconstruction formula (NRF) supplies the answer [2, e pluribus unum]. But the theoretical NRF uses an infinite sum, which must be truncated in some manner for practical application and generally performs poorly near the edge of a picture (for example), where the truncation becomes severe.

Polynomial spline techniques lack computational simplicity because they require a global solution: interpolated values at any point depend on data points that are far away [3]. Furthermore, if a higher-order spline is desired, a completely different solution must be applied.

The usual approach to purely local interpolation is to apply a low-order polynomial fit to the data in a short, sliding window consisting of an even number of points (e.g., $N = 2$, 4, or 6) to obtain interpolated values between the two central points. Figure 1 illustrates the problem for $N = 4$. Since only a few points are used, interpolations can be done very close to the edge of the picture. The discrete Fourier transform (DFT) interpolator described here is also local in nature, hence is computationally easy to apply. Furthermore, it uses the same formula for any number of data points and, depending on the data sampling rate, can give results superior to the local polynomial interpolators.

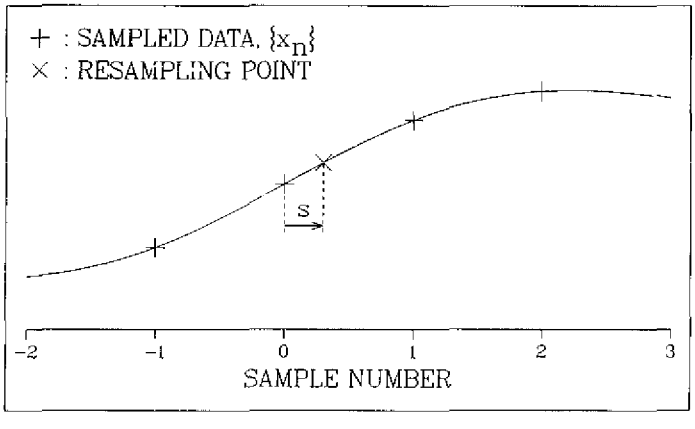

Fig. 1 — The solid curve represents the analog signal. An interpolated value at the shifted position s must be found from the sampled data and compared to the analog signal.

Manuscript approved February 25, 1992.

2. THE DFT INTERPOLATOR

Appendix A shows how the DFT and the Fourier shift theorem (FST) can be used to motivate a definition for an analog of the NRF for a limited number of data points. In one dimension, for an even number of points, the result is Eq. (A7), which is always applied with $n = N/2-1$; as with the local polynomials, interpolations are done only between the two central points in a sliding window, as indicated in Fig. 1. Putting this restriction into Eq. (A7) and appropriately relabeling the summation index to conform to Fig. 1, we have an interpolation formula to apply to a set of points $\{x_n\}, -N/2-1) \le n \le N/2$, to obtain an interpolated value for the point x_s, $0 < s < 1$:

$$x_s = \frac{1}{N} \sum_{(n=-N/2-1)}^{N/2} x_n \frac{\sin\,[\pi(s - n)]}{\tan\,[\pi(s - n)/N]}, \tag{1}$$

when N is even. The forms for odd N and for two dimensions are given by Eqs. (A8) and (A9). Note that since $N \cdot \tan[\pi(s-n)/N] \to \pi(s-n)$ as $N \to \infty$, Eq. (1) approaches the NRF in this limit.

Equation (1) has the form of a convolutional interpolation formula

$$x_s = \sum_{n=-(N/2-1)}^{N/2} x_n\, A_n(s), \tag{2}$$

where the interpolation coefficients $A_n(s)$ are obtained from an interpolation function $F(x)$ by

$$A_n(s) = F(n-s). \tag{3}$$

The interpolation function for DFT-N (N even) is obviously

$$F(x) = \frac{\sin(\pi x)}{N \tan(\pi x/N)}\, R_N(x), \tag{4}$$

where R_N is the rectangle function of width N centered at the origin; that is, $R_N(x) = 1$ for $|x| \le N/2$, $R_N(x) = 0$ otherwise. Equation (4), as applied in Eq. (1), now stands as a candidate interpolation function to be compared to conventional polynomial interpolation functions.

Previous applications of the DFT to the interpolation problem have centered on a "zero padding" method in frequency space due to Schafer and Rabiner [4] and recently investigated by Fraser [5]. This method is appropriate if a number of additional samples evenly spaced between existing samples is desired. An example would be a requirement for ten-times-denser resampling with nine equally spaced new samples between each pair of existing samples. But the technique is not well suited to generating a single new sample at an arbitrary location. The approach presented here *is* well suited to this problem and can be shown to give the same interpolated value as the zero padding technique by manipulating Fraser's Eq. (5). This can more readily be seen from his Eq. (6), which is derived from his Eq. (5), but his Eq. (6) unfortunately contains an error that prevents the calculated interpolated value from being a pure real number for nonintegral interpolation points (whereas his Eq. (5) does guarantee a real value). Once this error is corrected (see Appendix B of this report), Fraser's Eq. (6) is identical (in content) to this report's Eq. (A5).

Finally, an alternate form of the interpolation function that does not run the computational risk of a zero denominator is

$$F(x) = \frac{1}{N}\left[1 + \cos(\pi x) + 2\sum_{n=1}^{N/2-1}\cos(2n\pi x/N)\right] R_N(x) \tag{5}$$

for N even, and

$$F(x) = \frac{1}{N}\left[1 + 2\sum_{n=1}^{(N-1)/2}\cos(2n\pi x/N)\right] R_N(x) \tag{6}$$

for N odd.

Combining Eqs. (5) or (6) with Eqs. (2), (3), and (4) shows that the DFT interpolators are a special case of trigonometric interpolation [6]. They are equivalent to using the lowest-order trigonometric polynomial that can fit N sample points over the interval $(-N/2,N/2)$ when the sample points are at $-N/2 + 1, -N/2 + 2, \ldots, N/2$, for N even and at $-(N - 1)/2, -(N - 1)/2 + 1, \ldots, (N - 1)/2$ for N odd.

3. TESTING THE INTERPOLATOR

The coefficients of an N-point DFT are, of course, the coefficients of the first $N/2 + 1$ terms (including DC, see Section 4) in a Fourier series describing the data. If the data actually consisted of just those discrete frequencies used by the DFT, then the interpolation would be exact. To see that the interpolation formula works well on more realistic signals, even for small N ($N = 4,6,8$), it will be compared to polynomial interpolators applied to a representative point spread function (PSF).

A PSF is used as a test waveform because it is the most rapidly varying signal that will occur in most scenes and, therefore, the hardest to interpolate. (An example of a scene that can produce a more rapidly varying signal is a positive-contrast point source located very close to a negative-contrast point source, but such occurrences are rare.) The representative PSF chosen is a unit-height Gaussian, with standard deviation σ given in units of the sample spacing. Thus, by changing σ, we can see how well an interpolator works as a function of sampling rate.

The DFT interpolators are obtained from Eq. (1) with the appropriate value of N and will be compared to polynomials that use the same number of data points to perform an interpolation. Thus DFT-4, DFT-6, and DFT-8 will be compared to third-, fifth-, and seventh-order polynomials, respectively. The third order (cubic) interpolator used is that described by Wolberg [7] and extensively investigated by Park and Schowengerdt [8], with the parameter α chosen to be -0.5. This "image-independent optimal choice" [8] emphasizes high fidelity at low frequencies. The fifth-order (quintic) interpolator was derived by Schaum [9], using a more general method. The seventh-order interpolator will be standard Lagrange interpolation [1], denoted LF-8 (Lagrange function for $N = 8$). The formulas for these interpolators are given in Appendix C. Figure 2 plots the interpolation functions for all these interpolators.

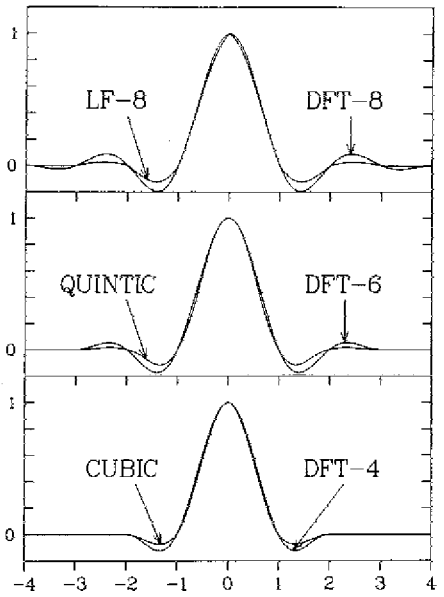

Fig. 2 — The DFT interpolation functions from Eq. (4) for N = 4, 6, 8. The polynomial interpolation functions obtained from Appendix 3 and Eq. (3) are included for comparison.

Figure 3 shows the results for $\sigma = 1$ and worst-phased sampling (samples miss the peak of the curve by the maximum amount). Figure 4 shows best-phased sampling (a sample occurs at the peak of the curve). The DFT interpolators fit the peak of a Gaussian better than the polynomials but tend to ring more at the foot of the curve. Errors calculated over the intervals shown in Figures 3 and 4 are given for various σ's in Table 1.

The following remarks about Table 1 emphasize worst-phased sampling because it is bound to occur at many places in real data. For $\sigma = 1.5$ (1.5 samples per standard deviation), the polynomials are superior to their DFT counterparts, but the differences are small; with well-sampled data, all interpolators perform well. For $\sigma = 0.5$, none give acceptable results (though the polynomials may be marginally adequate for best-phased sampling). The advantage of the DFT interpolators becomes apparent for the intermediate case of $\sigma = 1$. DFT-4, 6, and 8 give somewhat better results than the corresponding polynomials for root-mean-square (rms) errors and substantially better results for maximum errors. (DFT-10, not shown, yields a small further improvement, but the point of diminishing returns has been reached.)

The greater superiority of the DFT interpolators with respect to maximum errors than with respect to rms errors (as shown in Table 1) can be important. In the frame-differencing signal processor alluded to earlier, for example, reducing maximum errors will more quickly reduce the number of false alarms for a given threshold setting than reducing the rms error will.

The fact that DFT-7 gives worse results than DFT-6 is dealt with in Section 4. It is worth noting that if the NRF is truncated to ten points or fewer and normalized to have unit response at zero frequency, its performance is substantially worse than any of the interpolators shown here. It is also worth noting that $\sigma = 1$ is a somewhat critical value. If the PSF is sampled substantially more often than once per standard deviation, the choice of interpolator becomes unimportant. If it is sampled substantially less often, none perform well.

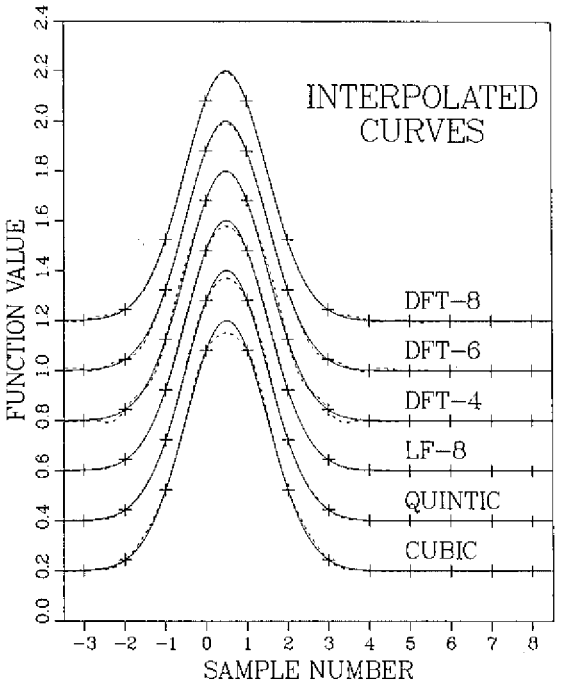

Fig. 3 — The solid curves are unit-height Gaussians with standard deviation equal to the sample spacing. The dashed lines show how well different interpolators perform in reproducing the Gaussians from the marked samples, with worst-phased sampling.

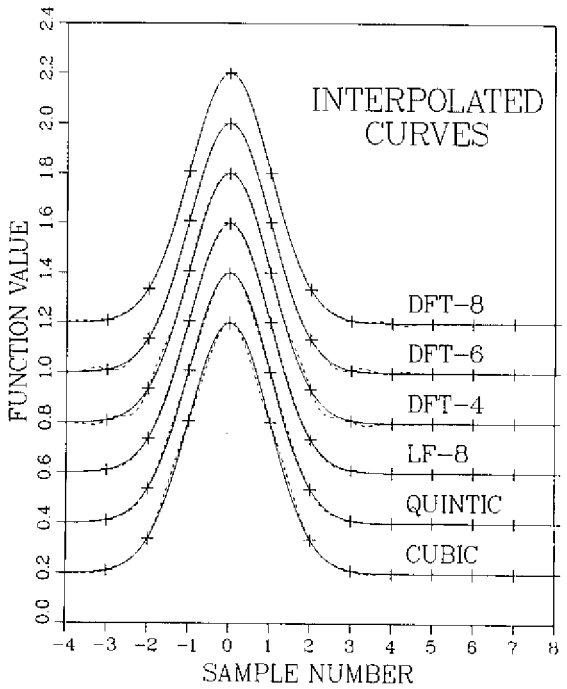

Fig. 4 — Same as Fig. 1, but with best-phased sampling. In this case, all the interpolators except cubic fit the Gaussian peak almost perfectly.

Table 1 — Interpolation Errors*

σ=	Interpolation Method	Worst Phase		Best Phase	
		max	rms	max	rms
0.5	Cubic	0.33	0.084	0.022	0.006
	Quintic	0.30	0.079	0.043	0.014
	LF-8	0.28	0.076	0.071	0.022
	DFT-4	0.28	0.077	0.066	0.021
	DFT-6	0.26	0.070	0.091	0.034
	DFT-7	0.23	0.064	0.13	0.063
	DFT-8	0.25	0.067	0.11	0.041
1	Cubic	0.050	0.013	0.032	0.012
	Quintic	0.030	0.008	0.020	0.007
	LF-8	0.022	0.006	0.013	0.005
	DFT-4	0.032	0.011	0.029	0.010
	DFT-6	0.016	0.006	0.014	0.005
	DFT-7	0.084	0.026	0.073	0.026
	DFT-8	0.010	0.004	0.008	0.003
1.5	Cubic	0.013	0.004	0.012	0.004
	Quintic	0.005	0.002	0.006	0.002
	LF-8	0.003	0.001	0.003	0.001
	DFT-4	0.021	0.009	0.021	0.009
	DFT-6	0.010	0.004	0.009	0.004
	DFT-7	0.076	0.030	0.072	0.030
	DFT-8	0.006	0.002	0.006	0.002

*Entries are maximum and root-mean-square errors over the intervals shown in Figs. 1 and 2 for Gaussians with the indicated standard deviations

Testing an interpolator on real data is very desirable but seldom possible; an interpolated value can be calculated, but the true value to which it should be compared is usually not available. To show that the superior performance of the DFT interpolators is not limited to Gaussian functions, they were compared to the quintic polynomial interpolator on accurately simulated data. This was done by using a scene simulator that produces images of the Earth as seen from a satellite, with 50- to 100-m resolution. The PSF for the simulated scene is close to, though not exactly, a Gaussian (but, of course, very little of the scene produces as hard-to-fit a signal as the PSF, so we should expect good interpolation performance for $\sigma < 1$). On scenes with little structure, all the interpolators performed well. On a highly structured scene (including an aerial view of a city) with $\sigma \approx 0.5$, all interpolators gave about equally poor results. With $\sigma \approx 0.75$, all gave reasonably good results, but the DFT interpolators performed better than the polynomials by about the margins shown in Table 1 for $\sigma = 1$ with worst-phased sampling. With $\sigma > 1$, all the interpolators performed well; quintic was slightly better than DFT-6, but not quite as good as DFT-8.

The lesson of the simulated imagery and of Table 1 is that the DFT interpolators give better performance at lower sampling rates (i.e., smaller σ) than the polynomials.

4. DISCUSSION

To see why the DFT-based interpolation formulas work well, we first consider in more detail the fact that interpolations are done only between the two center points of a sliding window.

Interpolation with the DFT and FST is sometimes done globally by transforming the entire data string, multiplying the Fourier coefficients by the appropriate phase factors (as is done with the g_k in Appendix 1), and transforming back to real space to obtain a set of interpolated samples [10,11]. This has the advantage of doing the interpolations between all data points simultaneously (as opposed to between the two central points only) but often gives disappointing results because of edge discontinuities.

Discontinuities at the edges of a data set are a common problem with Fourier techniques; if the data (or its derivatives!) do not wrap around smoothly, a Fourier analysis, which assumes cyclically repeating data, "sees" a discontinuity at the edge. Near any discontinuity, a Fourier series converges slowly and imperfectly, an effect known as Gibbs' phenomenon and covered in standard texts [12]. This problem is generally dealt with by various "windowing" methods (e.g., von Hann [13] or Hamming [14]). But these methods are imperfect (they impose their own structure on the image and are not band-limited) and impractical in a small data set. Figure 5, in which six- and seven-point DFT's are applied to a ramp, shows the effect of an edge discontinuity when interpolations are done between all the data points simultaneously. The six-point DFT can do accurate interpolations between the two central points; in this interval, the effect of the edge discontinuity is quite small. But interpolations for the seven-point DFT are strongly affected in all intervals. From this example, we expect a DFT interpolator to give much better results for an even number of points than for an odd number, and the comparison of DFT-7 with DFT-6 in Table 1 shows this to be the case.

Once a DFT-based interpolation formula is considered to be a standard interpolation process (applying only to the center interval of a sliding window), it becomes apparent that the main reason for its success is its good frequency response.

An N-point DFT describes its input data in terms of a few "allowed frequencies," i.e., frequencies for which an integral number of cycles fit exactly into the interval covered by N sample points. The maximum frequency contained in the DFT is $f_m = 1/2$, (i.e., one cycle per two samples), *regardless of the number of samples*. Thus even a few-point DFT directly measures the highest frequency available in

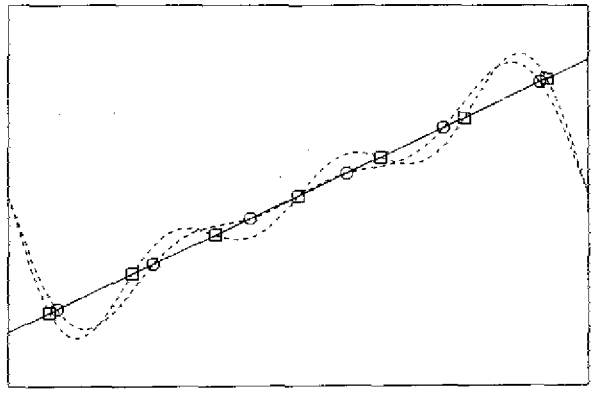

Fig. 5 — The dashed curves show the values interpolated between samples of a ramp by six-point (circles) and seven-point (squares) DFT interpolators. The former reproduces the central interval in the ramp very well, while the latter does not work well in any interval.

the sampled scene (the Nyquist frequency) as well as the lowest (the DC component). Of course, the few-point DFT has very poor frequency resolution; the difference between successive frequencies is $\Delta f = 1/N$, so there are only $N/2$ nonzero frequencies in the range from zero frequency to the Nyquist frequency. The DFT approximates intermediate frequencies (such as one half-way between two of the allowed frequencies) by a sum of the allowed frequencies.

The above remarks are for even N, which is the case of interest here. For the sake of completeness, the corresponding quantities for odd N are $\Delta f = 1/N$, as before, but $f_m = (1-1/N)/2$; therefore, the number of nonzero frequencies is $(N - 1)/2$. Thus a six-point and a seven-point DFT both contain three nonzero frequencies; the difference is that the six-point (or any even-point) DFT contains only the cosine phase of its highest frequency, while the seven-point (or any odd-point) DFT also contains the sine phase, as shown in Fig. 6.

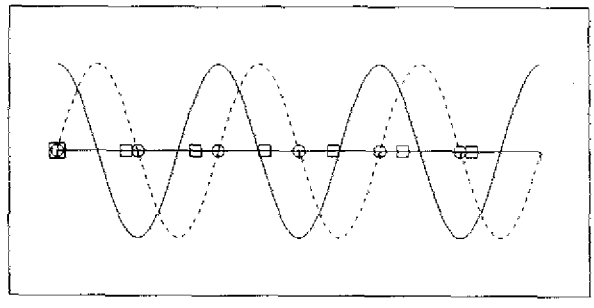

Fig. 6 — If three complete cycles of a sinusoid are covered by six samples (circles), only the cosine phase (solid line) yields nonzero values; the sine phase (dashed line) goes undetected. For seven samples (squares), both signals are detected. The "missing"point on the right belongs to the next six- or seven-point interval.

But even though a DFT interpolator has poor frequency resolution, it interpolates its allowed frequencies perfectly, and these frequencies are evenly distributed across the spectrum. This results in a more uniform frequency response than is possible with the polynomials that can be optimized to give better fits at some frequencies than others [8] but cannot reproduce any nonzero frequency exactly (since their outputs can only be a piece-wise continuous polynomial, not a true sinusoid). We will follow

standard practice in taking the Fourier transform of the interpolation function as a useful measure of the interpolator's frequency response (however, see below for further remarks on this point). The Fourier transform of $F(x)$ (which is easily found from Eq. (5)) is

$$G(k) = \frac{1}{2}\{\text{sinc } [N(k-1/2)] + \text{sinc } [N(k+1/2)]\} + \sum_{n=-(N/2-1)}^{N/2-1} \text{sinc } [N(k-n/N)], \quad (7)$$

for even N, where $\text{sinc}(x) \equiv \sin(\pi x)/(\pi x)$.

Figure 7 shows the Fourier transforms of the cubic, quintic, LF-8, and DFT-4, 6, and 8 interpolation functions (all the transforms are real and symmetric, so only the positive halves are shown). Note that the DFT interpolators have unit response at their allowed frequencies (i.e., unity at $f = 1/4$ for DFT-4, at 1/6 and 1/3 for DFT-6, 1/8, 1/4, and 3/8 for DFT-8), except at $f = 1/2$, where the response of 0.5 reflects the fact that only the cosine phase of this frequency is sampled. The maximum value at lower frequencies is 1.03. Also note that the DFT interpolators have broader-band response than the comparable polynomials. The ideal frequency response is the rectangle function R_1 (unity between $\pm 1/2$, zero otherwise; this is the Fourier transform of the NRF's sinc interpolation function). The DFT interpolation functions approach this limit as their order increases. The approach is especially rapid at low frequencies, which is important because the power spectral density of virtually all imagery has a much higher value at low frequencies than at high frequencies.

Interpreting the Fourier transform of the interpolation function as the frequency response of the interpolator is not as clear-cut as one would like. First, the term "frequency response" must be employed carefully for an interpolator because it is not well defined; an input sinusoid produces an output that is not a sinusoid. This is true for the polynomials at any nonzero frequency and for the DFTs at any frequency other than the allowed frequencies. Second, as shown in Appendix D, if an interpolator reproduces an input frequency exactly, then the Fourier transform will have the value unity at that frequency, but the converse is not true. For example, the parameters of the cubic interpolation function can be chosen so that its Fourier transform has unit value at some point in addition to zero [8], but since the result of an interpolation is a piece-wise cubic function, an input sinusoid at that frequency is not reproduced exactly.

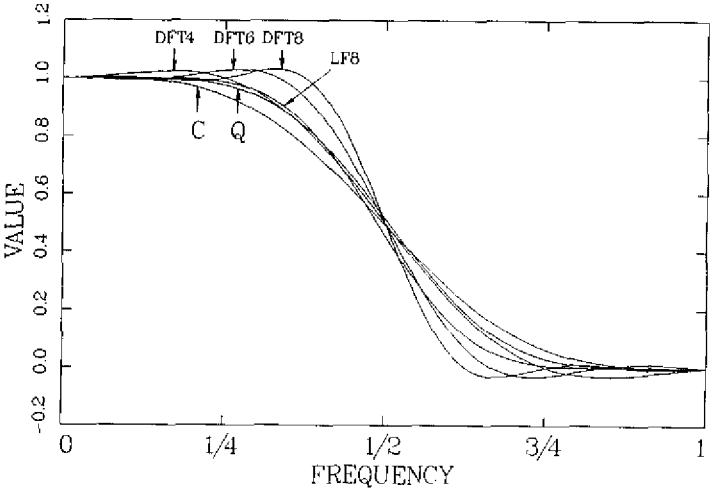

Fig. 7 — The Fourier transforms of the cubic (C), quintic (Q), LF-8, and DFT interpolation functions. The Nyquist frequency is $f = 1/2$.

5. CONCLUSION

DFT-based interpolators, when applied locally to the center of a sliding window, have been shown to be a valuable addition to the signal processor's repertoire. On highly structured data, they can extend good interpolation performance to lower data sampling rates than is possible with local polynomial interpolators.

REFERENCES

[1] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes* (Cambridge University Press, New York, 1986), Ch. 3.

[2] A.V. Oppenheim, and R.W. Schafer, *Digital Signal Processing* (Prentice-Hall, Inc., Englewood, New Jersey, 1975), p. 29.

[3] W.H. Press *et. al.*, *op. cit.*, pp. 86-89.

[4] R.W. Schafer, and L.R. Rabiner, "A Digital Signal Processing Approach to Interpolation," *Proc. IEEE*, **61**, 692-702 (1973).

[5] D. Fraser, "Interpolation by the FFT Revisited — An Experimental Investigation," *IEEE Trans. Acoustics, Speech, and Sig. Proc.* **37**, 665-675 (1989).

[6] P.J. Davis, *Interpolation and Approximation* (Dover Publications, NY, 1975), pp. 29, 38.

[7] G. Wolberg, *Digital Image Warping* (IEEE Computer Society Press, Washington, 1990), pp. 129-131.

[8] S.K. Park, and R.A. Schowengerdt, "Image Reconstruction by Parametric Cubic Convolution," *Computer Vision, Graphics, and Image Processing* **23**, 258-272 (1983).

[9] A.P. Schaum, Naval Research Laboratory, Washington, D.C., private communication.

[10] W.H. Haas, and C.S. Lindquist, "A Synthesis of Frequency Domain Filters for Time Delay Estimation," *IEEE Trans. Acous., Speech, Sig. Proc.*, **21**, 540, (1981).

[11] S. Holm, "FFT Pruning Applied to Time Domain Interpolation and Peak Localization," *IEEE Trans. Acoustics, Speech, Sig. Proc.*, **35**, 1777 (1987).

[12] W. Rogosinski, *Fourier Series* (Chelsea Publishing Company, New York, 1959), p. 135.

[13] R.W. Hamming, *Digital Filters* (Prentice-Hall, Inc., Englewood, New Jersey, 1983), p. 102.

[14] W.K. Pratt, *Digital Image Processing*, (John Wiley & Sons, Inc., New York, 1978), p. 294.

APPENDIX A

Derivation of Interpolation Formula

The form of the Nyquist reconstruction formula for a limited number of data points is exhibited below (Eqs. (A7), (A8), or (A9)). A sketch of the derivation is given for a one-dimensional discrete Fourier transform (DFT) for even N. The changes in the derivation for odd N are then indicated, and the generalization to two dimensions is presented.

For a set of N equally spaced samples $\{x_n\}$, $0 \le n \le N-1$ from a continuous function $f(u)$ (i.e., $x_n = f(n)$), the coefficients of the corresponding DFT are (following the conventions of Oppenheim and Schafer [A1])

$$g_k = \sum_{n=0}^{N-1} x_n \, e^{-j2\pi kn/N} \quad 0 \le k \le N-1 \tag{A1}$$

Thus,

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} g_k \, e^{j2\pi kn/N}. \tag{A2}$$

Recognizing that $g_{N-k} = g^*_k$ for $k \ge 1$ (and therefore, that for even N, $g_{N/2} = g^*_{N/2}$ = real number), we can write this as

$$x_n = \frac{1}{N} \left[g_0 + g_{N/2}(-1)^n + 2Re \sum_{k=1}^{N/2-1} g_k \, e^{j2\pi kn/N} \right]. \tag{A3}$$

The Fourier shift theorem (FST) [A2] states that if $F(k)$ is the (continuous) Fourier transform of $f(x)$, then the Fourier transform of $f(x + s)$ is $F(k)\exp(iks)$. The interpolation problem consists of finding a value for $f(n + s)$ given $\{x_n\}$. When the DFT is expressed in the form Eq. (A3), comparison of the FST and the interpolation problem suggests the following *definition* of a new set of DFT coefficients:

$$g_k = g_k \, e^{j2\pi ks/N} \quad \text{for } k \le N/2-1,$$

and

$$g'_{N/2} = g_{N/2} \, Re(e^{j\pi s}) = g_{N/2} \cos(\pi s), \tag{A4}$$

where s, $0 \le s \le 1$ denotes the shift. The definition of $g'_{n/2}$ is chosen to assure that the coefficient of the highest frequency in the DFT is real, as it must be for N even, if it is to be the DFT of a real-valued (as opposed to complex-valued) sequence of data. By defining $\{x_{n+s}\}$ to be the sequence of real numbers that will result from the inverse DFT of the coefficients defined in Eq. (A4) and substituting their values in terms of the g_k, we have a candidate expression for an interpolated sample:

$$x_{n+s} = \frac{1}{N}\left[g_0 + g_{N/2}(-1)^n \cos(\pi s) + 2Re \sum_{k=1}^{N/2-1} g_k\, e^{j2\pi k(n+s)/N}\right]. \tag{A5}$$

Whether x_{n+s} closely approximates $f(n + s)$ remains to be seen (for $n = N/2 - 1$, it does, see text). We can now substitute for the g's from (Eq. (A1)), use the facts that $(-1)^{n-m}\cos(\pi s) = \cos[\pi(n + s - m)]$ and

$$\sum_{k=1}^{N/2-1} u^k = u\frac{1 - u^{N/2-1}}{1 - u} = u^{N/4}\,\frac{u^{(N-2)/4} - u^{-(N-2)/4}}{u^{1/2} - u^{-1/2}}, \tag{A6}$$

with $u = \exp[j2\pi(n + s - m)/N]$ to obtain, for even N,

$$x_{n+s} = \frac{1}{N}\sum_{m=0}^{N-1} x_m\,\frac{\sin[\pi(n+s-m)]}{\tan[\pi(n+s-m)/N]}$$

$$= \frac{\sin(\pi s)}{N}\sum_{m=0}^{N-1} x_m\,(-1)^{n-m}\cot[\pi(n+s-m)/N]. \tag{A7}$$

For odd N, $N/2$-1 is replaced by $(N - 1)/2$ in Eq. (A3) and g_0 is the only term that appears outside the summation. The rest of the derivation is unchanged. The result for odd N is

$$x_{n+s} = \frac{\sin(\pi s)}{N}\sum_{m=0}^{N-1} x_m(-1)^{n-m}\csc[\pi(n+s-m)/N]. \tag{A8}$$

The generalization to two dimensions is obviously

$$x_{n+s,m+r} = \frac{\sin(\pi s)\,\sin(\pi r)}{NM}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} x_{k,l}(-1)^{n+m-k-l}\cot[\pi(n+s-k)/N]\cot[\pi(m+r-l)/M]. \tag{A9}$$

Here the one-dimensional interpolation is applied sequentially in each direction. If either (both) of N and M are odd, then the corresponding cotangent function(s) is (are) replaced by cosecant(s).

REFERENCES

[A1] A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing* (Prentice-Hall Inc., Englewood, NJ, 1975), p. 89.

[A2] R.N. Bracewell, *The Fourier Transform and Its Applications* (McGraw-Hill Book Company, New York, 1978), pp. 121-2.

APPENDIX B

Correction of Earlier Work

This appendix corrects an error in Fraser's 1989 paper on fast Fourier transform interpolation [B1]; it must be read in conjunction with that paper to be comprehensible.

N will be taken to be even (for N odd, Fraser's paper contains no error), so Fraser's $N^{-}/2$ can simply be replaced by $N/2$. To see that Fraser's Eq. (6) contains an error, the special case $U(N/2)$ = real number, all other $U(k) = 0$, will be considered. Then Fraser's Eq. (5), which correctly gives the values interpolated from the original sequence, which is denoted by $u(i)$, $i = 0,1,2,....N - 1$, becomes

$$\begin{aligned} v(i') &= V(N/2)\exp[j2\pi i'(N/2)/L] + V(L-N/2)\exp[j2\pi i'(L-N/2)/L] \\ &= 0.5 * U(N/2)\ [\exp(j\pi i'/M) + \exp(-j\pi i'/M)] \\ &= U(N/2)\ \cos(\pi i'/M),\ i' = 0,1,2,....L-1, \end{aligned} \tag{B1}$$

where Fraser's Eqs. (3) and (4) and $L = M \cdot N$ have been used. In Eq. (B1), the interpolated values are always real. However, for the same special case, Fraser's Eq. (6), which is ostensibly obtained from his Eq. (5) by letting $i'/M \rightarrow x$ (that is, letting i'/M become a continuous variable), gives

$$\begin{aligned} v(Mx) &= U(N/2)\exp(j\pi x) \\ &= U(N/2)\exp(j\pi i'/M) \text{ for } x = i'/M \end{aligned} \tag{B2}$$

which is not real for most i'. Thus, there must be an error in Fraser's Eq. (6). After putting $x = i'/M$ in Fraser's Eq. (5), a few straight-forward steps lead to

$$\begin{aligned} v(Mx) = &\sum_{k=0}^{N/2-1} U(k)\exp(j2\pi kx/N) + \sum_{k=1}^{N/2-1} U(k)^*\ \exp(-j2\pi kx/N) \\ &+ 0.5U(N/2)[\exp(j\pi x) + \exp(-j\pi x)], \end{aligned} \tag{B3}$$

where use has been made of the fact that $U(N/2)$ must be real because it is the $N/2$ coefficient of an N-term DFT (with N even) of real data. This is the correct version of Fraser's Eq. (6). It can be seen to be identical in content (though not in the $1/N$ normalization convention) to this report's Eq. (A5) by setting $x = n + s$.

REFERENCE

[B1] D. Fraser, "Interpolation by the FFT Revisited--An Experimental Investigation," *IEEE Trans. Acous., Speech, and Sig. Proc.* **37**, 665-75 (1989).

APPENDIX C

Polynomial Interpolation Formulas

The formulas used for the polynomial interpolators are given below.

Cubic

$$x_s = \frac{1}{2} \sum_{n=-1}^{2} x_n A_n,$$

where

$$A_{-1} = s(1-s)^2$$

$$A_0 = 2 - s^2(5-3s)$$

$$A_1 = s(1 + 4s - 3s^2)$$

$$A_2 = - s^2(1-s)$$

Quintic

$$x_s = \frac{1}{24} \sum_{n=-2}^{3} x_n B_n,$$

$$B_{-2} = s(2-s) - s^3(9 - 13s + 5s^2)$$

$$B_{-1} = - 16s(1-s) + s^3(39 - 64s + 25s^2)$$

$$B_0 = 24 - 30s^2 - s^3(70 - 126s + 50s^2)$$

$$B_1 = 16s(1 + s) + s^3 (66 - 124s + 50s^2)$$

$$B_2 = - s(2+s) - s^3(33 - 61s + 25s^2)$$

$$B_3 = s^3(7 - 12s + 5^2)$$

Standard Lagrange interpolation [C1] consists of fitting an (N - 1) order polynomial through N data points. Specialized to the problem of interest here, this yields LF - 8:

$$x_s = \sum_{n=-3}^{4} x_n C_n, \text{ with } C_n = \prod_{\substack{m=-3 \\ m \neq n}}^{4} \frac{(s-m)}{(n-m)}.$$

REFERENCE

[C1] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes* (Cambridge University Press, New York, 1986), Ch. 3.

APPENDIX D

Interpolation Errors for Pure Sinusoids

First we define two error terms E_c and E_s by the equations

$$\sum_{m=-(N/2-1)}^{N/2} \cos(2\pi km)\ F(s-m) = \cos(2\pi ks) + E_c(k,s) \tag{D1}$$

and

$$\sum_{m=-(N/2-1)}^{N/2} \sin(2\pi km)\ F(s-m) = \sin(2\pi ks) + E_s(k,s), \tag{D2}$$

where $0 < s < 1$. The left-hand sides of Eqs. (D1) and (D2), where $F(x)$ is the interpolation function from Eq. (3), will be recognized from Eq. (1) as the interpolated values of cosine and sine inputs, respectively. Thus, E_c and E_s are the errors resulting from the interpolation. In what follows, the first equality is the Fourier transform of the interpolation function (which is an even function [e.g., Eq. (4)], so its Fourier transform has no imaginary component). The second equality is obtained by the change of variables $x \rightarrow m - s$ for each value of m. The third equality is then found by expanding $\cos[2\pi k(s-m)]$, using $F(m - s) = F(s - m)$, and substituting from Eqs. (D1) and (D2).

$$G(k) = \int_{-N/2}^{N/2} F(x)\ \cos\ (2\pi kx)dx \tag{D3}$$

$$= \int_0^1 \sum_{m=-(N/2-1)}^{N/2} \cos[2\pi k(s-m)]\ F(s-m)ds \tag{D4}$$

$$= 1 + \int_0^1 \left(E_c(k,s)\ \cos(2\pi ks) + E_s(k,s)\ \sin(2\pi ks)\right)ds. \tag{D5}$$

This is the desired result: if $E_C = E_S = 0$ for some k (e.g., for a DFT interpolator at an allowed frequency), then $G(k) = 1$. But $G(k)$ can be unity with E_C, $E_S \neq 0$, if the integral in Eq. (D5) is zero. There is, unfortunately, no simple relation between the error of the interpolator and the nonunity of the Fourier transform of the interpolation function.